Automated Theorem Proving

Lecture 12: Superposition

Prof. Dr. Jasmin Blanchette
based on slides by Dr. Uwe Waldmann

Winter Term 2024/25

Part 5: Superposition

First-order calculi considered so far:

Resolution: for first-order clauses without equality.

(Unfailing) Knuth–Bendix completion: for unit equations.

Goal:

Combine the ideas of ordered resolution (overlap maximal literals in a clause) and Knuth–Bendix completion (overlap maximal sides of equations) to get a calculus for equational clauses.

5.1 Recapitulation

First-order logic:

Atom: either $P(s_1, \dots, s_m)$ with $P \in \Pi$ or $s \approx t$.

Literal: atom or negated atom.

Clause: (possibly empty) disjunction of literals
(all variables implicitly universally quantified).

Recapitulation

Refutational theorem proving:

For refutational theorem proving, it suffices to consider sets of clauses: every first-order formula F can be translated into a set of clauses N such that F is unsatisfiable if and only if N is unsatisfiable.

In the nonequational case, unsatisfiability can for instance be checked using the (ordered) resolution calculus.

Recapitulation

(Ordered) resolution: inference rules:

Ground case: Nonground case:

Resolution:

$$\frac{D' \lor A \qquad C' \lor \neg A}{D' \lor C'} \qquad \frac{D' \lor A \qquad C' \lor \neg A'}{(D' \lor C')\sigma}$$

where $\sigma = \mathrm{mgu}(A, A')$.

Factoring:

$$\frac{C' \lor A \lor A}{C' \lor A} \qquad \frac{C' \lor A \lor A'}{(C' \lor A)\sigma}$$

where $\sigma = \mathrm{mgu}(A, A')$.

Recapitulation

Ordering restrictions:

Let $\succ$ be a well-founded and total ordering on ground atoms.

Literal ordering $\succ_L$:
compares literals by comparing lexicographically first the respective atoms using $\succ$ and then their polarities (negative $>$ positive).

Clause ordering $\succ_C$:
compares clauses by comparing their multisets of literals using the multiset extension of $\succ_L$.

Recapitulation

Ordering restrictions (ground case):

Inference are necessary only if the following conditions are satisfied:

- The left premise of a "Resolution" inference is not larger than or equal to the right premise.
- The literals that are involved in the inferences ($[\neg]\,A$) are maximal in the respective clauses
(strictly maximal for the left premise of "Resolution").

Recapitulation

Ordering restrictions (nonground case):

Define the atom ordering $\succ$ also for nonground atoms.

Need stability under substitutions: $A \succ B$ implies $A\sigma \succ B\sigma$.

Note: $\succ$ cannot be total on nonground atoms.

For literals involved in inferences, we have the same maximality requirements as in the ground case.

Recapitulation

Resolution is (even with ordering restrictions) refutationally complete:

Dynamic view of refutational completeness:

If N is unsatisfiable ($N \models \bot$), then *fair* derivations from N produce $\bot$.

Static view of refutational completeness:

If N is *saturated*, then N is unsatisfiable if and only if $\bot \in N$.

Recapitulation

Proving refutational completeness for the ground case:

We have to show:

> If N is saturated (i.e., if sufficiently many inferences have been computed), and $\bot \notin N$, then N has a model.

Recapitulation

Constructing a candidate interpretation:

Suppose that N be saturated and $\bot \notin N$.
We inspect all clauses in N in ascending order and construct a sequence of Herbrand interpretations
(starting with the empty interpretation—all atoms are false).

If a clause C is false in the current interpretation, and has a positive and strictly maximal literal A, then extend the current interpretation such that C becomes true: add A to the current interpretation.
(Then C is called *productive*.)

Otherwise, leave the current interpretation unchanged.

Recapitulation

The sequence of interpretations has the following properties:

(1) If an atom is true in some interpretation, then it remains true in all future interpretations.

(2) If a clause is true at the time where it is inspected, then it remains true in all future interpretations.

(3) If a clause $C = C' \vee A$ is productive, then C remains true and C' remains false in all future interpretations.

Show by induction: If N is saturated and $\bot \notin N$, then every clause in N is either true at the time when it is inspected or productive.

Recapitulation

Note:
For the induction proof, it is not necessary that the conclusion of an inference is contained in N.
It is sufficient that it is redundant w.r.t. N.

N is called *saturated up to redundancy* if the conclusion of every inference from clauses in $N \setminus \mathit{Red}(N)$ is contained in $N \cup \mathit{Red}(N)$.

Recapitulation

Proving refutational completeness for the nonground case:

If $C_i\theta$ is a ground instance of the clause C_i for $i \in \{0, \ldots, n\}$ and

$$\frac{C_n \quad \cdots \quad C_1}{C_0}$$

and

$$\frac{C_n\theta, \ldots, C_1\theta}{C_0\theta}$$

are inferences, then the latter inference is called a ground instance of the former.

Recapitulation

For a set N of clauses, let $G_\Sigma(N)$ be the set of all ground instances of clauses in N.

Construct the interpretation from the set $G_\Sigma(N)$ of all ground instances of clauses in N:

N is saturated and does not contain $\bot$
$\Rightarrow$ $G_\Sigma(N)$ is saturated and does not contain $\bot$
$\Rightarrow$ $G_\Sigma(N)$ has a Herbrand model I
$\Rightarrow$ I is a model of N.

Recapitulation

It is possible to encode an arbitrary predicate P using a function f_P and a new constant *true*:

$$\begin{array}{rcr} P(t_1,\dots,t_n) & \leadsto & f_P(t_1,\dots,t_n) \approx \mathit{true} \\ \neg\, P(t_1,\dots,t_n) & \leadsto & \neg\, f_P(t_1,\dots,t_n) \approx \mathit{true} \end{array}$$

In equational logic it is therefore sufficient to consider the case that $\Pi = \emptyset$, i.e., equality is the only predicate symbol.

Abbreviation: $s \not\approx t$ instead of $\neg\, s \approx t$.

5.2 The Superposition Calculus—Informally

Conventions:

From now on: $\Pi = \emptyset$ (equality is the only predicate).

Inference rules are to be read modulo symmetry of the equality symbol.

We will first explain the ideas and motivations behind the superposition calculus and its completeness proof. Precise definitions will be given later.

The Superposition Calculus—Informally

Ground inference rules:

Pos. Superposition: $$\frac{D' \lor t \approx t' \qquad C' \lor s[t] \approx s'}{D' \lor C' \lor s[t'] \approx s'}$$

Neg. Superposition: $$\frac{D' \lor t \approx t' \qquad C' \lor s[t] \not\approx s'}{D' \lor C' \lor s[t'] \not\approx s'}$$

Equality Resolution: $$\frac{C' \lor s \not\approx s}{C'}$$

(Note: We will need one further inference rule.)

The Superposition Calculus—Informally

Ordering wishlist:

Like in resolution, we want to perform only inferences between (strictly) maximal literals.

Like in completion, we want to perform only inferences between (strictly) maximal sides of literals.

Like in resolution, in inferences with two premises, the left premise should not be larger than the right one.

Like in resolution and completion, the conclusion should then be smaller than the larger premise.

The ordering should be total on ground literals.

The Superposition Calculus—Informally

Consequences:

The literal ordering must depend primarily on the larger term of an equation.

As in the resolution case, negative literals must be slightly larger than the corresponding positive literals.

Additionally, we need the following property:
If $s \succ t \succ u$, then $s \not\approx u$ must be larger than $s \approx t$.
In other words, we must compare first the larger term, then the polarity, and finally the smaller term.

The Superposition Calculus—Informally

The following construction has the required properties:

Let $\succ$ be a *reduction ordering that is total on ground terms.*

To a positive literal $s \approx t$, we assign the multiset $\{s, t\}$,
to a negative literal $s \not\approx t$ the multiset $\{s, s, t, t\}$.
The literal ordering $\succ_{\mathrm{L}}$ compares these multisets using the multiset extension of $\succ$.

The clause ordering $\succ_{\mathrm{C}}$ compares clauses by comparing their multisets of literals using the multiset extension of $\succ_{\mathrm{L}}$.

The Superposition Calculus—Informally

Constructing a candidate interpretation:

We want to use roughly the same ideas as in the completeness proof for resolution.

However, a Herbrand interpretation does not work for equality:
The equality symbol $\approx$ must be interpreted by equality in the interpretation.

The Superposition Calculus—Informally

Solution: Productive clauses contribute ground rewrite rules to a TRS R.

The interpretation has the universe $\mathrm{T}_\Sigma(\emptyset)/R = \mathrm{T}_\Sigma(\emptyset)/{\approx_R}$;
a ground atom $s \approx t$ holds in the interpretation if and only if $s \approx_R t$
if and only if $s \leftrightarrow_R^* t$.

We will construct R in such a way that it is terminating and confluent.
In this case, $s \approx_R t$ if and only if $s \downarrow_R t$.

The Superposition Calculus—Informally

One problem:

The completeness proof for the resolution calculus depends on the following property:

> If $C = C' \vee A$ with a strictly maximal and positive literal A is false in the current interpretation, then adding A to the current interpretation cannot make any literal of C' true.

This property does not hold for superposition:

> Let $b \succ c \succ d$.
> Assume that the current rewrite system (representing the current interpretation) contains the rule $c \rightarrow d$.
> Now consider the clause $b \approx d \vee b \approx c$.

The Superposition Calculus—Informally

We need a further inference rule to deal with clauses of this kind, either the "Merging Paramodulation" rule of Bachmair and Ganzinger or the following "Equality Factoring" rule due to Nieuwenhuis:

Equality Factoring: $$\frac{C' \vee s \approx t' \vee s \approx t}{C' \vee t \not\approx t' \vee s \approx t'}$$

The Superposition Calculus—Informally

What do the nonground versions of the inference rules for superposition look like?

Main idea as in the resolution calculus:

Replace identity by unifiability.

Apply the mgu to the resulting clause.

In the ordering restrictions, use $\not\preceq$ instead of $\succ$.

The Superposition Calculus—Informally

However:

As in Knuth–Bendix completion, we do not want to consider overlaps at or below a variable position.

Consequence: There are inferences between ground instances $D\theta$ and $C\theta$ of clauses D and C which are *not* ground instances of inferences between D and C.

Such inferences have to be treated in a special way in the completeness proof.

5.3 The Superposition Calculus—Formally

Until now, we have seen most of the ideas behind the superposition calculus and its completeness proof.

We will now start again from the beginning giving precise definitions and some proofs.

The Superposition Calculus—Formally

Inference rules (part 1):

Pos. Superposition:

$$\frac{D' \lor t \approx t' \qquad C' \lor s[u] \approx s'}{(D' \lor C' \lor s[t'] \approx s')\sigma}$$

where $\sigma = \mathrm{mgu}(t, u)$ and
u is not a variable.

Neg. Superposition:

$$\frac{D' \lor t \approx t' \qquad C' \lor s[u] \not\approx s'}{(D' \lor C' \lor s[t'] \not\approx s')\sigma}$$

where $\sigma = \mathrm{mgu}(t, u)$ and
u is not a variable.

The Superposition Calculus—Formally

Inference rules (part 2):

Equality Resolution:
$$\frac{C' \vee s \not\approx s'}{C'\sigma}$$

where $\sigma = \mathrm{mgu}(s, s')$.

Equality Factoring:
$$\frac{C' \vee s' \approx t' \vee s \approx t}{(C' \vee t \not\approx t' \vee s \approx t')\sigma}$$

where $\sigma = \mathrm{mgu}(s, s')$.

The Superposition Calculus—Formally

Theorem 5.3.1:

All inference rules of the superposition calculus are sound, i.e., for every rule

$$\frac{C_n \quad \cdots \quad C_1}{C_0}$$

we have $\{C_1, \dots, C_n\} \models C_0$.

Proof:

Omitted. $\square$

The Superposition Calculus—Formally

Orderings:

Let $\succ$ be a *reduction ordering that is total on ground terms.*

To a positive literal $s \approx t$, we assign the multiset $\{s, t\}$,
to a negative literal $s \not\approx t$ the multiset $\{s, s, t, t\}$.
The literal ordering $\succ_{\mathrm{L}}$ compares these multisets using the multiset extension of $\succ$.

The clause ordering $\succ_{\mathrm{C}}$ compares clauses by comparing their multisets of literals using the multiset extension of $\succ_{\mathrm{L}}$.

The Superposition Calculus—Formally

Inferences have to be computed only if the following ordering restrictions are satisfied (after applying the unifier to the premises):

– In superposition inferences, the left premise is not greater than or equal to the right one.

– The last literal in each premise is maximal in the respective premise, i.e., there exists no greater literal
(strictly maximal for positive literals in superposition inferences, i.e., there exists no greater or equal literal).

– In these literals, the lhs is neither smaller than nor equal to the rhs (except in equality resolution inferences).

The Superposition Calculus—Formally

A ground clause C is called redundant w.r.t. a set of ground clauses N if it follows from clauses in N that are smaller than C.

A clause is redundant w.r.t. a set of clauses N if all its ground instances are redundant w.r.t. $G_\Sigma(N)$.

The set of all clauses that are redundant w.r.t. N is denoted by $\mathit{Red}(N)$.

N is called saturated up to redundancy if the conclusion of every inference from clauses in $N \setminus \mathit{Red}(N)$ is contained in $N \cup \mathit{Red}(N)$.

The Superposition Calculus: Example

We consider the clause set (Bentkamp et al., *CACM* 2023)

$$
\begin{aligned}
x \approx zero \vee div(one, x) \approx inv(x) \quad & (1)\\
pi \not\approx zero \quad & (2)\\
abs(div(one, pi)) \not\approx abs(inv(pi)) \quad & (3)
\end{aligned}
$$

using an lpo with the precedence $abs > div > inv > pi > one > zero$.

From (1) and (3), we obtain via "Negative Superposition"
$pi \approx zero \vee abs(inv(pi)) \not\approx abs(inv(pi))$ (4).
From (4), we obtain via "Equality Resolution" $pi \approx zero$ (5).
From (5) and (2), we obtain via "Negative Superposition" $zero \not\approx zero$ (6).
From (6), we obtain via "Equality Resolution" the empty clause.